



Paper Type: Research Paper

## Optimizing Identity and Access Management through 1D-SCNN-Based Anomaly Detection

Prabhadevi Cheruku<sup>1,\*</sup> , Vb Narasimha<sup>1</sup>

<sup>1</sup> Department of Computer Science and Engineering, Univeristy College of Engineering, Osmania University, Hyderabad, India; dasariprabhadevi@gmail.com; vbnarasimha@gmail.com.

### Citation:

Received: 10 August 2024  
Revised: 13 October 2024  
Accepted: 20 November 2024

Cheruku, P, & Narasimha, Vb. (2024). Optimizing identity and access management through 1D-SCNN-based anomaly detection. *Journal of applied research on industrial engineering*, 11(4), 574-592.


### Abstract


Identity and Access Management (IAM) systems are critical in the ever-evolving digital landscape as legacy security methods fall short against modern cyber threats. This study proposes a fast and accurate anomaly detection method named 1D-Separable Convolutional Neural Networks (1D-SCNN), effectively detecting abnormal identity access or credential abuse. This method utilizes deep learning to analyze user activity and access habits from a one-dimensional structure, leveraging the benefits of 1D-SCNN, such as lower computational cost and higher model efficiency. The proposed model employs a 1D-SCNN architecture customized for efficient anomaly detection in IAM systems. It uses separable convolutions to handle one-dimensional input data, reducing the number of parameters and required computation. The architecture includes layers such as Leaky ReLU and ELU for activation, MaxPooling for down-sampling features, Dropout for regulating overfitting, and a Flatten layer for classification. This configuration allows the model to learn from historical user engagement data and identify anomalous behavior patterns, which are strong indicators of security threats. The study results highlight the value of advanced deep learning techniques in cybersecurity and provide a roadmap for integrating 1D-SCNN within IAM systems to enhance security in digital environments. Finally, in experiments on an extensive data set, the proposed model outperformed by achieving an impressive accuracy of 96%.

**Keywords:** Identity and access management, 1D-Separable convolutional neural networks, Leaky ReLU, ELU.

## 1 | Introduction

Rising security threats have made the strong Identity And Access Management (IAM) in enterprises a vital necessity [1]. These systems are central in managing who gets access to which company resources, making sure only allowed individuals can interact with different data or systems. The complexity of cyber-attacks and the dynamic nature of modern IT environments have impacted traditional IAM solutions less [2]. Therefore,

 Corresponding Author: dasariprabhadevi@gmail.com

 <https://doi.org/10.22105/jarie.2024.472705.1657>

 Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

there is an urgent want for more sophisticated and flexible methods to protect digital identities and efficiently control access rights. Machine learning (ML) has been investigated for anomaly detection in IAM systems because it can learn and evolve when new threats come [1]. Traditionally, security systems work off a series of manually written rules and heuristics, whereas ML-based approaches can process large amounts of data to determine patterns and deviations that imply malicious activity [3]. This is particularly useful in areas like IAM where abnormal activity, e.g., unlikely login times, regions, or trends for a certain user, may indicate an unauthorized entity trying to access their account/stepping outside of intended use [4].

The increasing frequency of even more complicated identity incidents makes it necessary for a better approach to enhance IAM security [2]. In addition, there is a rise in the use of cloud services, remote work, and mobile devices, which has expanded the potential attack landscape with conventional perimeter-based security approaches. These ML methods for anomaly detection can prove to be an effective way to improve the existing IAM frameworks, thereby enabling teams in enterprises to proactively understand and respond early enough against potential threats that could disrupt them.

This makes ML a simple response to the ever-evolving threat landscape within IAM systems. The upcoming ML algorithms based on the standard level of typical activity against every user or function within the system will analyze and learn these behaviors regarding user behavior and access patterns [5]. This baseline detects anomalies that could suggest a break-in attempt, internal security breach, or compromised login details. This includes a request from an unknown location or one to access resources identified as sensitive during non-business hours, for example. This technique improves security and adapts to the dynamic nature and cautions from authorized users, lowering the rate of false alarms and improving user satisfaction.

Using ML for anomaly detection in IAM is also relevant to the broader trend towards zero trust security paradigms, where networks are not trusted by default, and individuals must authenticate each use case [6]. ML algorithms enhance this by providing a sophisticated system to continuously assess the risk associated with each access attempt based on context and behavior. This ability is important and helps detect and take action against advanced threats that are often invisible. Organizations deploy ML to fortify their IAM systems in this boundary-less environment, paving the way for enhanced digital landscape security and stability. IAM solutions are essential in today's digital world for protecting private information and controlling who has access to what. Handling large-scale systems or high-frequency access data may be particularly challenging for current approaches when balancing efficiency, scalability, and accuracy. However, current IAM anomaly detection methods have problems with scalability, False Positive (FP) rates, and real-time insights, particularly in complex and dynamic access patterns environments. Traditional models also fail to use new machine-learning capabilities when spotting complex abnormalities and possible security breaches.

The main contribution of the paper is:

- I. Designing the 1D-Separable Convolutional Neural Networks (1D-SCNN) effectively detects abnormal identity access or credential misuse.
- II. The architecture includes layers such as Leaky ReLU and ELU for activation, MaxPooling for down-sampling features, Dropout for regulating overfitting, and a Flatten layer for classification.
- III. Experimental outcomes have been analyzed, and the suggested 1D-SCNN model increases the performance ratio, accuracy, precision, recall, and F1-score ratio compared to existing models.

The rest of the paper is prearranged as follows: Section 2 deliberates the literature survey, Section 3 proposes the 1D-SCNN model, Section 4 discusses the results and discussion, and Section 5 concludes the research paper.

## 2 | Literature

Nassif et al. [7] conducted a Systematic Literature Review (SLR) on ML models for anomaly identification in different applications. The models are analyzed in terms of various aspects, such as anomaly detection

applications, ML methodologies employed, and performance indicators for the ML model. We identify 290 research publications from year-2000 through 2020 that serve as a basis for ML methods in anomaly detection. Given our dataset from research publications, we have tagged 43 different anomaly detection applications. Furthermore, 29 different machine-learning models were identified for anomaly detection.

Xu et al. [8] proposed a forensic analysis method using different algorithms to scrutinize and purify the data to detect intrusions, anomalies, etc. Synthetic Minority Over-sampling Techniques (SMOTE) and mutual information are employed to improve the training set's quality. This is done using automated ML to determine which method with optimized hyperparameters performs best in classifying the data. This method was used in studies specifically selected for computational efficiency to find the best parameters without requiring time-consuming hyper-parameter optimization during each run-time test.

Demertzis et al. [9] introduced an innovative Blockchain Security Architecture in the literature. It is constructed in a way that assists in connecting Industrial IoT devices endorsed by the standard Industry 4.0. Its architecture is based on DL Smart Contracts. The smart contracts proposed would enforce a bilateral traffic management agreement in computer code, calling attention to anomalous behavior through trained Deep Autoencoder Neural Networks (DANN). Such design allows the development of a secure distributed platform with no single central authority to manage and execute related transactions within critical infrastructure networks.

Liu et al. [10] offered a theoretical survey of ML solutions for recognizing and counterfeiting malicious IoT devices, specifically defaulting to passive surveillance agents or network operators. The author can divide the identification and recognition of IoT devices into 4 groups: 1) device-specific pattern detection, 2) DL-based device detection, 3) unsupervised device detection, and 4) irregular device identification. In the meantime, other ML tools that help achieve this goal were discussed. The enabling technologies are feature engineering, learning algorithms on network traffic traces and incremental learning, wireless signals, and anomaly identification.

Bagaa et al. [11] introduced an advanced security architecture built using ML that resolves the increasingly hazardous IoT environments with autonomy. This framework incorporates Software Defined Networking (SDN) and Network Function Virtualization (NFV) to alleviate multiple security risks. The integrated monitoring agent and AI-based response agent consist of machine-learning models that belong to one category: network pattern analysis (loosely used for anomaly-based intrusion identification in IoT systems).

Quatrini et al. [12] designed this experiment to verify the necessity of process visible identification on anomaly detection as indicated. The method benefits from the decision forests algorithm, widely used for anomaly detection in industrial data sources. A lesser-known forest-based technique called decision jungle has not been applied to an industry-standard problem. A real case study in the pharmaceutical industry confirms its efficacy by using an anomaly identification method utilizing a 10-month dataset with 16 progression variables from a granulation progression.

Sarker et al. [13] extensively investigated IoT security intelligence, utilizing machine and deep learning models to interpret raw data that optimizes defense mechanisms for the wide spectrum of cyber threats targeting IoT devices. We reveal our analysis and stress the research topics for future scopes of related work.

De Carvalho Bertoli et al. [14] introduced a structure for the AB-TRAP called framework, which helps to address operational issues while allowing implementation of authorities and coverage that applies to actual network data. It follows a well-understood sequence of events: 1) the production of the attack datasets, 2) bonafide datasets, 3) the training of the ML model, 4) the realization of the model, and 5) the performance assessment of realized models post-deployment. The author tested the AB-TRAP, an open-source tool developed to identify TCP port scanning attacks in both Local Networking (LAN) and global networking (internet).

Ola Salman et al. [15] introduced a framework for determining IoT devices and identifying malicious communication. This method collects features per network flow to infer the origin of generating traffic,

identify what it produces from traffic, and determine whether there are any threats to a network. It achieves this by distributing intelligence to the edges of networks.

Waheed et al. [16] provided an extensive literature review of the work published in the last decade, from 2008 to 2019, on IoT's privacy and security challenges with ML algorithms and BC approaches. The author starts by providing an overview of various privacy and security risks stated in the domain of IoT over the past decades. In the following, we apply ML algorithms and BC techniques to classify the literature dealing with security & privacy initiatives in an IoT ecosystem.

Aboukadri et al. [1] suggested ML in IAM systems. There has been a direct correlation between the development of IAM and the rise of AI integration as a critical path toward making IAM more successful. The integration of ML approaches to strengthen IAM is the subject of this review, which zeroes in on the three main operations of IAM authorization, authentication, and auditing. To tackle the core concerns about ML's impact on IAM operations, the author presents a detailed description of IAM in unified layered reference models. This model will emphasize the functions of authorization, authentication, and auditing, particularly monitoring. Ultimately, this study considerably enhances the IAM landscape by offering a thorough understanding of the revolutionary function of ML. The study plans to use ML to enhance IAM systems' performance, security, and effectiveness in response to critical concerns.

Cremonezi et al. [17] proposed Identity management for the IoT. This study offers an in-depth analysis of IdMs, including the current methods, key principles, and problems involved in creating an IdM for IoT applications. After introducing the reader to the IdM concept and its goals, elements, and models, the author draws parallels between these ideas and the traits of the IoT. The author then explores the main obstacles to modifying IdMs for the Internet of Things (IoT). The author reviews the literature that suggests ways to fix at least one part of IdMs in light of current problems. In addition, the study brings attention to outstanding problems in this area, such as authentication for IoT devices, scalability of identity provisioning, and worries about the administration and performance of authorization procedures.

Dabbaghi Varnosfaderani et al. [18] recommended the two self-attention-based models (SmartSSO) for automated account linkage in federated identity management. To discover associated user accounts, SmartSSO modifies two models that rely on self-attention to create new representations based on user patterns in a latent space with fewer dimensions. Over six months, 50,000 users contributed over a million samples to the production data set, and the trained models reached a hit-precision accuracy of over 98%.

Algarni [19] discussed the deep learning based parking lot allotment to the vehicles. Technology and sensors designed specifically for parking garages allow for more efficient parking by directing cars to available spots. These sensors and technology are not yet operating in the open parking lot. This study assesses how well they work in open parking lots by examining the literature on intelligent parking sensors, technologies, and applications. Further investigation into deep learning and multi-agent systems will be needed before open parking lots can benefit from smart parking solutions.

Vladislav [20] deliberated the leakage detection in water pipes. The primary objective is to identify instances of water distribution network leaks during water separation. Our solution to this issue is using SWLD, or Smart Water Leakage Detection, in pipes. The SWLD monitors the tank's water level and activates the pump when the water level drops. The initial component of the system is an alarm that utilizes GSM technology to communicate with the owner using Short Message Service (SMS). The system includes the essentials, such as sensors, a GSM module, and an Arduino. Second, it's the portion that controls the pump; it uses a mobile app for Android.

Hariri et al. [21] presented the cumulative and deep learning algorithms for diagnosis of heart failure. Utilizing integrated classification algorithms, this research set out to analyze data from 451 samples representing 13 features of internal medicine patients at risk of heart failure to identify patterns of cardiovascular disorders. To improve classifiers and make them more accurate, it is necessary to choose features and assess the aspects that have an impact. As a result, the author looked into the Gini index elements. Decision trees, neural

networks, and other cumulative approaches like random forests, gradient boosting, and the innovative DL technique were used in the classification phase. According to a comparison, deep learning improved the neural network's presentation and findings with 95.33% accuracy, 95.77% illness class accuracy, and 94.74% health class accuracy.

Shitharth et al. [22] introduced the Federated learning optimization for the computational blockchain procedure with offloading analysis to improve security. The suggested method's main strength is its use of a blockchain-based offloading methodology for data processing, which guarantees the utmost confidentiality for all data. The use of data weights in load balancing, together with parametric assessments performed in real-time to ascertain the consistency of every data monitored with IoT, is a result of a problem approach built concerning clusters. Results that were investigated using a five-scenario procedure show that offloading analysis with blockchain is safer, leading to 89% more accurate data processing for all IoT applications.

Khodaverdian et al. [23] investigated the CNN and Gated Recurrent Units (GRU) for energy-aware resource allocation in virtual machine translation. The workload of Virtual Machines (VM) in the microsoft azure dataset is either delay-sensitive (interactive) or delay-insensitive, and the dataset is labeled accordingly. The dataset is imbalanced in sample distribution, with the bulk of samples labeled delay-insensitive. To tackle the difficulty, this article utilizes the Synthetic Minority Oversampling Techniques (SMOTE) technique. Empirical findings demonstrated that their suggested model outperformed other models with an accuracy of 95.18%.

Khodaverdian et al. [24] examined the shallow Deep Neural Networks (DNN) for selecting migration candidate VM to Reduce Energy Consumption. This research presents a method for diagnosing a virtual machine's latency sensitivity using a mix of CNN and GRU. The goal is to select the best candidate VM for migration. The workload of Microsoft Azure VM was used as a dataset to evaluate the suggested model. The suggested model outperforms current models in terms of classification accuracy when choosing the VM that should be migrated, as shown by the empirical data.

Safa et al. [25] analyzed the prediction of mental health using social media. Data gathering, feature extraction, and prediction algorithms are the pillars upon which this study rests. On top of that, the author looks back at several recent studies that have investigated various aspects of candidate profiles and evaluation techniques. The author then examines current and future developments in experimental auto-detection frameworks for disease identification, debating several elements of their evolution. In the future, the presented methodologies may make diseases simpler to treat, augment screening protocols, and detect at-risk individuals via large-scale social media monitoring.

Kuang et al. [26] suggested the hybrid DL Method for sentiment analysis in product reviews. The author of this paper suggests a new way to analyze product evaluations for sentiment utilizing a mixture of traditional methods and DNNs. The main emphasis is on sentiment classification. The Recursive Neural Network (RNN) technique is used to classify sentiment. The author uses a resampling strategy to make the dataset more representative of the minority class while lowering the majority class's sample size to fix the uneven distribution of negative and positive observations in social network data. The four product categories that comprise Amazon's data set, clothing, automobiles, luxury items, and home appliances, are used to assess our method. The suggested technique outperforms the existing sentiment analysis for product evaluations, tested in a digital marketing setting. Not only that, but the attention-based RNN algorithm achieves a 5% improvement over the baseline RNN.

Darwish [27] proposed a data-driven DL method for the remaining useful life of rolling bearings. To improve the accuracy of RUL predictions for rolling bearings, this research introduces a DL model that uses a CNN, LSTM, and attention mechanisms. To extract features, CNN first evaluates input data in the temporal domain. Then, an attention mechanism is used to align the output and input series according to the semantics or content of the input series, after which two LSTM layers are used to record complex temporal correlations and generate more abstract data depictions. In the end, a Fully Connected (FC) layer makes the predictions. The author compared the suggested model's performance to other deep learning models and used the IEEE



PHM 2012 Challenge dataset to assess the model's usefulness. The proposed CNN-ALSTM model outperformed all other models tested, suggesting it is a solid option for RUL prediction of rolling bearings.

Elmasry et al. [28] recommended using deep learning to differentiate Authentic News from Fabrications. In this research, the author proposes a new kind of deep learning model called CNN-DNN, which combines CNN with DNN. The author tested the suggested model on the 72,134 news stories that comprise the WELFake dataset. Among all the articles, 35,028 are true news, and 37,106 are fake news. Regarding accuracy, the suggested model came out on top with 0.9732, while the LSTM model came out on the bottom with 0.960. The primary objective was to create a detection mechanism to effectively limit the spread of misinformation.

Eskandari [29] presented the Python-Based Information Theoretical Multi-Label Feature Selection Library (PyIT-MLFS). New algorithms can be more easily developed with the help of the PyITMLFS package. When creating algorithms for multi-label feature selection, this is the first open-source toolkit covering all the bases. It also has high-level interfaces that users may use to test and evaluate various currently used algorithms.

Mahalakshmi et al. [30] introduced the q-rung orthopair multi-fuzzy systems for advancing cybersecurity strategies for multinational corporations. This research aims to improve the ability to distinguish between such sets by presenting two new distance measures optimized for Q-Rung Ortho-Pair MFSs (q-ROMkFSs) of dimension  $k$ . This study presents a procedural development in decision-making procedures by expanding its application to Multi-Attribute Decision Making (MADM) using score functions relevant to q-ROMkFSs. A comparison with current MADM approaches reveals the effectiveness of the suggested measures, highlighting the offered approach's superiority. This research deepens our understanding of q-ROMFSs and their theoretical foundations, which can help multinational corporations formulate more effective cybersecurity strategies. To provide a thorough tool for handling cybersecurity issues, the research uses the Hamming and Euclidean distance measurements as standards, and it also develops a score and an accuracy function.

Mahmoud Ismail and Amal F.Abd El-Gawad [31] suggested the revisiting zero-trust security for iot. this article delves into the zero-trust framework's core principles. Then, it examines how they apply to the IoT, highlighting the need for encryption, micro-segmentation, continuous authentication, and stringent access restrictions. Likewise covered are the ever-changing dangers that iot systems confront and how well zero-trust principles protect sensitive data, ensure the integrity of devices, and strengthen the system. In addition, the paper emphasizes important points about the difficulties and factors to consider while implementing zero-trust security in various IoT infrastructures.

Alanazi and Alrashdi [32] investigated the CNN-LSTM for anomaly detection in smart agriculture systems. The author simulates possible DDoS attack scenarios by rigorously training and testing our model using two separate sensor input datasets. Some important measures to evaluate the model's performance are comparing the call, F1-score, and detection accuracy. The data show that our technique is successful, with a remarkable 99.7 percent accuracy rate in anomaly identification. In the long run, this study improves the sustainability and dependability of farming by paving the way for more effectual attack and anomaly identification methods for network-edge intelligent agriculture models.

Rasinojehdehi and Najafi [33] analyzed the Data Envelopment Analysis (DEA) and Strategic Assessment of Network Security. The author shows how to use DEA in a real-world setting by considering 10 separate networks as DMUs and evaluating their security. Computer network security may be categorized into four levels: terribly insecure, insecure, safe, and very safe based on the performance metrics that are derived. Reducing the amount of inputs and outputs using Principal Component Analysis (PCA) allows us to maximize the discriminating power of DEA. This guarantees both the accuracy of our assessment and the suitability of the quantity of DMUs for the analysis. The discriminating power of DEA may be maintained by generally increasing the number of DMUs by a factor of three above the total of inputs and outputs. This study adds to the existing body of knowledge by developing a thorough and efficient technique for assessing and

categorizing computer network security via integrating DEA and PCA. The results provide important insights into better-protecting networks from cyber attacks.

When planning the development of multi-purpose transmission in smart grids, Saberi and Hatef [34] used Pareto power evolutionary algorithms based on multi-objective particle pool optimization (SPEA2-MOPSO) to analyze the situation according to operator priorities (cost or risk). The system's load and security resources should be taken into consideration. The numerical verification of the suggested methods is performed on the modified IEEE RTS 24-bus and 118-bus systems. The simulation findings show that the suggested method can coordinate the best course of action for planning and disaster recovery while providing information about different types of hazards.

Zaferani et al. [35] suggested an automated personality assessment using the Big Five Inventory (BFI). We use an auto-encoder as a nonlinear feature learning approach to extract and choose relevant features for the classification. A stop criteria based on maximal separation ability in binary classes is used to find a saddle point since an auto-encoder is unable to extract adequate classification alone. The findings show that for the majority of personality characteristics, nonlinear features improve the classification outcomes. Additionally, to characterize the uncertainty originating from mental states and influence the classification outcomes using the retrieved features, we employ an adaptive neuro-fuzzy inference system classification. The findings of four qualities show a notable increase in the classification results on the SSPNet Speaker Personality dataset. These byproducts confirm that the speech signal contains ambiguity.

To handle sequential data, many conventional models use complicated architectures like LSTMs; however, this method uses depthwise separable convolutions, greatly reducing computing costs without sacrificing accuracy. Although LSTM-based approaches sometimes have longer inference times and need more training resources, the 1D-SCNN outperforms them with an accuracy of 94.2% and a better ROC-AUC score of 0.972. Separable convolutions are important for real-time anomaly identification and improve processing performance for massive access management datasets. This efficiency shows the uniqueness of our method in conjunction with competitive performance measures.

### 3| Proposed Method

The goal of the approach involving anomaly detection in IAM through 1D-SCNN is to secure digital systems. Based on user activity and access patterns, this approach leverages deep learning to identify anomalous behaviors associated with unauthorized or malicious activities more accurately. This approach is based on a 1D-SCNN, or custom convolutional neural network extended to process only sequential input via separable convolutions efficiently. This technology increases the ability to investigate access logs and user interactions in an easily understandable, one-dimensional manner, providing faster discovery of anomalies with great accuracy. This strategy exploits the unique benefits of 1D-SCNN, such as reduced computation overhead and model performance boost, to deliver an adaptive proactive mechanism that ensures sensitive data confidentiality and incident-free security applications. It also emphasizes the need for cutting-edge deep learning tools to stay ahead of unrelenting cyber threats.

#### 3.1| Separable Convolutional Neural Network

Separable CNNs are a type of CNN architecture designed to increase computational efficiency while maintaining the same level of performance in applications like image recognition, object detection, and segmentation. Efficiency is achieved by separating the filtering process by breaking down the usual convolution process into two distinct operations: depthwise convolutions and pointwise convolutions.

Depthwise convolution: in depthwise convolution, every filter convolves only across individual input channels; unlike a conventional convolution layer, their weights are independent of the number of output layers(outputs produced) and input channel (inputs given). In depthwise convolution, the corresponding filter is convolved channel-wise, i.e., each input channel is filtered with a different learnable 2D kernel that is

respective to its original dimensions. There will be  $C$  convolution processes, one per filter if there are  $C$  input channels and  $K$  filters. This operator for each input channel independently collects the spatial information.

Pointwise convolution: after depthwise convolution, pointwise convolution is done. A  $1 \times 1$  convolution that acts on a pixel location of all channels independently. This stage combines spatial information from depthwise convolution and creates new features via linear combinations. The depthwise convolution effectively produces a projection in the new feature space.

Separable convolutions confer several advantages:

- I. Decreased computational complexity: the overall number of computations used is significantly less than that for a traditional convolution since the convolution process separates into depthwise and pointwise operations. Depthwise convolution decreases the computational costs by factors of  $C$  (number of input channels), and pointwise convolution similarly decreases it by a factor  $K$  ( $K$  represents the number of output channels).
- II. Fewer parameters: separable convolutions contain fewer parameters than normal ones, reducing the risk of overfitting, especially in scenarios with insufficient training data.
- III. Enhanced efficiency: due to its lower computational complexity and reduced number of parameters, separable convolutions are more efficient in training and deployment. This qualifies them well for scarce resources like mobile devices or embedded systems.
- IV. In cases where intricate spatial relationships need to be captured, separable convolutions might not consistently achieve the same level of performance as standard convolutions. Such cases should carefully balance the trade-off between computational efficiency and speed when performing.
- V. Separable CNNs balance computation efficiency and performance by replacing the regular convolution with depthwise and pointwise convolutions. Thus, they are well suited to scenarios with lightweight computing resources or high performance to this slight architectural tweak, but they do not sacrifice.

### 3.2 | Proposed SCNN Architecture

#### Separable Conv 1D

Fig. 1 shows the Separable Convolutional 1D (also known as Separable Conv 1D), a technique used to deal with one-dimensional input data such as time series or sequential data within the context of CNNs. Classic convolutional operation enables a filter to move across the input data and, at each location, calculates the dot product between those filter weights and that part of input values. The play sets two separate separable processes, depthwise convolution, and pointwise convolution, which are separately called Separable Conv 1D.



Fig. 1. Proposed method architecture.



Each input channel is processed separately using a different filter in the depthwise convolution phase. This methodology documents the locational relationships of cells by individual channels, respectively. Then, in the pointwise convolution, a 1x1 Convolution layer combines all the depth-wise output channels to get the final output.

A combination of depthwise and pointwise convolutions has reduced the parameters needed compared to conventional convolutional layers while still keeping important information about the data. This allows learning more efficient models with fewer applicable processing demands in resource-limited settings and for large datasets.

The first convolutional layer has 32 3-size filters, while succeeding layers include 64-128-, 256-size, and depthwise separable convolutions to decrease processing. After every convolutional block, dimensionality reduction is achieved by applying max-pooling layers with a pool size of 2. The model employs a 128-neureceptor FC layer with softmax activation for classification. The Adam optimizer trains the network using 50 epochs, a batch size of 64, and a learning rate of 0.001. For model assessment, this study uses the categorical cross-entropy loss function, and to avoid overfitting, this study uses dropout layers with a 0.5 dropout rate.

### Leaky ReLU

Leaky ReLU is an improved version of the Rectified Linear Unit (ReLU), and it got its name from how negative inputs handle 0 values. The traditional ReLU activation function makes the negative inputs 0 and positive inputs equal. The ReLU activation function is widely used due to its simplicity and effectiveness in solving the notorious vanishing gradient problem, but it has a downside known as the "dying ReLU" phenomenon.

When training, if a neuron receives enough negative inputs, it may become inactive and keep its output to zero forever. That way, during backpropagation, a zero gradient is pushed back through the network, and the weights are not updated, which ultimately stops further learning. The leaky ReLU technique solves that issue by adding some positive slope for negative inputs instead of doing output to zero.

This little slope helps to handle the dying ReLU problem (factors in which gradient=0 and hold back learning from training properly) by allowing a small quantity of gradient to move backward throughout the network even when inputs are negative. The mathematical definition of Leaky ReLU is  $f(x)=\max(ax, x)$ , where 'a' is a tiny constant usually set to a very small value such as 0.01. Since leaky ReLU allows a small gradient when the unit is not active, it can be used to alleviate problems related to dead neurons and adds variance into activation in the network.

This little incline ensures the neurons being given negative input are still part of the gradient in backpropagation, which is crucial for the learning process, particularly in complex neural networks. Sometimes, Leaky ReLU has appeared to replace the standard ReLU for DNN, especially when a dying ReLU problem knocks off a big chunk of neurons.

Leaky ReLU represents an additional hyperparameter, which needs to be tuned properly like other parameters that could complicate the model fitting and optimization process. Still, Leaky ReLU is important to deep learning researchers as it can mitigate the issues with regular ReLUs.

### MaxPooling 1D

MaxPooling 1D is a common example in deep learning, specifically with CNNs applied to tasks like image recognition, audio classification, and natural language processing. CNNs are one of the most common neural network architectures used for image data but can also be applied to time series or text sequences after certain modifications.

MaxPooling 1D is a way to down-sample incoming data by reducing its dimensionality. 1D MaxPooling works across the time dimension of the input sequence.

The function of MaxPooling is that the most important aspects present in input data are picked up & become shorter, which makes it more computationally efficient, thereby preventing overfitting. MaxPooling involves sliding a fixed window over the input sequence and selecting the max value inside each window. It represents the most salient feature in that frame.

MaxPooling always selects the highest value; this is vital to retain important elements, and unwanted items are processed. It helps in remembering 2 extra attributes of the input sequence and shortening its length. This would allow a deeper network to focus more on the global structure of the input while using the MaxPooling layer in return for location-invariance features. This process pools the maximum value of every window in which some features are located, and this significantly reduces how much of a shift in sequence can affect the outcome because only the highest value is kept for each window. This helps CNNs to learn features invariant of changes in rotation, size, and variations in the input data, thus making it more generalizable.

### **Dropout**

Dropout is a relatively standard regularization method for neural networks, specifically deep learning models trained on data sets that may be paralyzed using methods such as classification or regression. The idea is to prevent overfitting, which means the model loses its ability to generalize over a new dataset but falls back into remembering specific samples in the training data set. Dropout results are obtained using a model with better generalization because neurons are less correlated, allowing them to learn more robust features.

The dropout role is to randomly turn off a fraction of the neurons by setting them to zero during each training epoch. However, some neurons will be temporarily dropped out or ignored during forward pass and backpropagation. Dropout helps overcome this overfitting problem by preventing the model from relying too closely on a particular set of neurons.

It is essentially a form of ensemble learning, where different groups of neurons are trained independently to create a stronger and more adaptive model. Dropout will randomly select a set of neurons with a likelihood indicated as the hyperparameter that can be defined at any time during training (for instance, 0.2 means that 20% of neurons are excluded).

There may be probability variations depending on the model's complexity and dataset. All neurons adjust their output in the forward pass to reflect this new average predicted output. The gradients of dropped-out neurons are not passed back during the backward pass, so co-adaptation of neurons is avoided.

Dropout has a big advantage by being as simple and efficient as possible. This can easily be fitted into various neural network protocols without introducing much computational overhead. It can be implemented anywhere between layers (input, hidden, and finally, output layer), most commonly performed in the middle (hidden) layers.

Given a specific architecture and dataset, this is especially convenient for fine control of the impact of regularization from dropout. During the inference phase after training, dropout is generally turned off so that all neurons contribute to making predictions. This ensures that the model scales up to reach near-perfect predictions without noise.

Dropout is a very useful regularization technique that reduces overfitting, allows the model to generalize, and helps improve the robustness of such models, hence increasing the performance system at multiple ML tasks.

### **ELU**

On the other hand, ELU works a little better than ReLU; It can output negative numbers, while ReLU outputs all smaller values to 0 because it can produce negative numbers, unlike those that set any negative values to zero. As a result, this method addresses the dying ReLU problem by allowing neurons to have negative activations, thus ensuring that they can be updated during backpropagation to participate in the learning process.

The ELU function can also produce a non-zero gradient at all input values, alleviating the vanishing gradient problem. The fact that the gradient is not 0 (all values are greater than zero) means gradients can be back-propagated through this operation to all weights in a deep neural network, which helps train models faster and more reliably.

The ELU function has the property of detecting activations near zero, and it is possible that this can lead to better convergence properties for a neural network that may be faster on some problems. Because the ELU function is more complex than easier activation functions such as ReLU, the ELU may have been computationally expensive because it involves exponential operations. However, the gains in performance, especially for complex structures, usually outweigh computational costs.

### Flatten

The flattened layer is necessary for deep learning, especially when using neural networks focused on image recognition or natural language process tasks. Its main purpose is to flatten the data, converting multidimensional arrays or tensors into one-dimensional arrays. It is critical for most neural networks because it converts the data between convolutional layers and a dense FC layer.

As image processing is used to analyze images, CNNs are neural networks that use input data as a tensor. While a grayscale image can be shown as a 2D array of pixel values, it would be a 3D array with height, width, and channels (e.g., RGB) for color images. Flatten these arrays before feeding them into the dense layers.

However, the problem is that convolutional layers are not built to do classification (or regression); they process input data and extract features from it; meanwhile, dense layers handle the final task of a network. A flattening layer works as a "middle ground" between them. After training on the input data and extracting relevant features, the output of the last convolution layer is presented to a flattened layer, which converts this into a 1D Array.

The flattened data format retains the spatial correlations between features retrieved from convolutional layers and prepares it for further input into dense networks. Different types of layers absorb various ranks and shapes; if a flattened layer is part of the model, each element in the input tensor to the layer must be rearranged into one continuous string while keeping invariant data. The Flatten layer flattens input tensors of dimensions (channels, batch\_size, width, height) into a one-dimensional array with shape(channels\*batch\_size\*width\*height). This converts the input into a single continuous vector that allows subsequent dense layers to analyze the data with an appreciation for intricate patterns and linkages.

CNN layers with separable convolutions have been demonstrated to be effective at identifying localized patterns in the input data, patterns that could relate to certain aspects or behaviors of user access. The Leaky ReLU or the ELU activation function offers non-linearity to the network, helping it understand complex relations and patterns in an input. MaxPooling reduces the size of feature maps and learns to find out which are important features. Dropout layers help prevent overfitting by randomly turning off fractions of input units during training.

The flattened layer adapts the feature map to one-dimensional vectors, which will now be utilized as input for dense networks for further processing or classification. The architecture is also implemented to extract important features from IAM data efficiently.

The novelty of the presented model is in integrating various approaches with distinct architectural changes applied to the proposed Separable Convolutional Neural Network (SCNN) architecture, even after deciding on several relevant independent parameters.

- I. The new way to handle 1D input data, such as time series or sequences, is using Separable Convolutional 1D. This operation separates the convolution process into depthwise and pointwise convolutions, reduces computational load and parameter count, and keeps necessary data information. This is most useful for resource-deprived environments or when the data source has a high volume of information.

- II. Also, Leaky ReLU can be an activation function that solves the "dying ReLU" problem in traditional ReLU activation functions. Leaky ReLU fixes the dying and maintains some activity in the neurons because of a little positive slope for negative inputs, which makes it possible to learn much more complex functions using backprop.
- III. MaxPooling 1D reduces the input sequence size by identifying important features so that fewer computational resources are needed, improving the generalization feature.
- IV. Dropout layers reduce overfitting by randomly turning off a certain percentage of neurons when training the model, which enables one to get even more robust qualities plus boosts generalization.
- V. The Exponential Linear Unit (ELU) function is used instead of the other regular ReLU form. The neuron directly goes through zero and is not just prone to dying out but also helps with negative activation, solves the gradient vanishing problem and improves training efficiency & more competitive convergence properties.

The flattened layer is critical to converting the 2D output from convolutional layers into some form that can become an input for FC Dense or Neural Network classifiers.

The novelty of this work lies in the extensive integration and use of multiple different methods within the SCNN architecture to meaningfully extract important features from input data while addressing computational complexity as well as overfitting and vanishing gradients. More details can help improve the capacity and robustness of a model for diverse machine-learning tasks, especially in scenarios like small resources or large datasets.

## 4 | Experimental Results

This section summarises the results obtained from simulations made with the suggested method. This investigation uses a network anomaly detection dataset from Kaggle [36]. The dataset was preprocessed using the method provided. R2L (remote to local) and U2R (user to root) are two potential cybersecurity threats directed at one of the four areas/segments by focusing on their specific inherent weaknesses in computer systems. Those attacks symbolize different stages of unauthorized access and violation of the security state within a system. Network security is taking center stage due to the exponential rise in the number of applications using networks and the total number of users connected. All Computer systems include vulnerabilities in security that are expensive and technically challenging for manufacturers to fix.

Consequently, Intrusion Detection Systems (IDSs) are playing an increasingly crucial role as specialized devices that can identify network threats and abnormalities. Anomaly-based and misuse-based detection approaches have dominated intrusion detection research for quite some time. Although commercial solutions often choose misuse-based detection for its high accuracy and predictability, academic research often views anomaly detection as a more powerful technique because of its theoretical capacity to handle new types of attacks. Several ML techniques have achieved an extremely high detection rate of 98% while maintaining a false alarm rate of 1%, according to a comprehensive review of the current research trend in anomaly detection. There is little indication of using anomaly detection methodologies in the state-of-the-art IDS Solutions and commercial solutions, and practitioners still see it as an emerging technology. Many studies in anomaly detection have sought to explain this discrepancy by investigating a wide range of factors, including learning and detection methodologies, training and testing datasets, and assessment tools. *Table 1* shows the experimental setup.

**Table 1. Experimental setup.**

Component	Description
Data split	70% Training, 15% Validation, 15% Testing
Number of layers	4 convolutional layers + 1 FC layer
Convolutional filters	32, 64, 128, 256 filters in successive layers
Filter size	3 in all layers
Pooling layers	Max pooling, pool size = 2
Activation function	ReLU (after each convolution)
Batch size	64
Optimizer	Adam (learning rate = 0.001)
Loss function	Categorical cross-entropy
Epochs	50
Regularization	Dropout (rate = 0.5), L2 regularization
Statistical tests	t-test, Wilcoxon signed-rank test (to compare models)

## R2L

An R2L (or remote-to-local) attack is a cyber-attack where an attacker without access to the system collaborates with another user of the server machine. Typically, the targeted local computer helps exploit server software or protocol vulnerabilities. R2L attacks are different from other forms of attack in that they come across the network (like a remote login) and bypass any defenses on site. Such attacks leverage vulnerabilities in the authentication mechanism, authorization systems, or account management functions to sign on as regular users and directly view data files.

Hence, the password attack is an R2L Attack where attackers use weak or default known passwords to grant themselves unauthorized access. When this happens, attackers can try multiple usernames and passwords repeatedly until they get lucky with one that grants them access. Another method is to exploit insecurity in network protocols or services running on the target machine, such as buffer overflow vulnerabilities within network daemons. R2L attacks pose a huge risk against either an enterprise or the individuals, as they have the power within them to cause data breaches and illegal entry of important systems, along with setting off service interruptions. To mitigate the risk of R2L attacks, organizations should improve authentication methods and regularly update software and system settings; they also need to gain visibility into user/network activity for signs of unauthorized access.

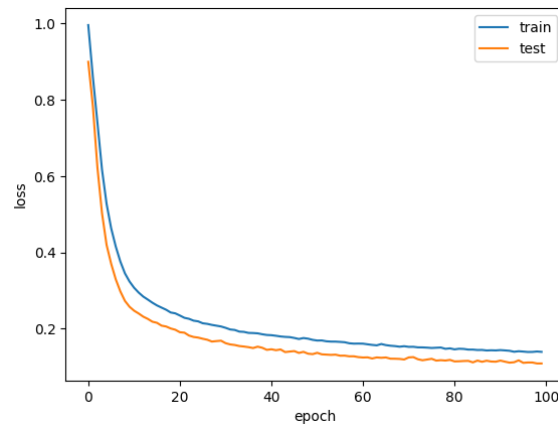
## U2R

U2R attack is a type of cyber-attack when the attacker, who has limited access to the system, tries to increase his rights and obtain root/administrator privileges. Such an attack commonly uses software or OS vulnerabilities to attain enhanced privileges. Once a hacker has root access, he is in full control of the system, thus potentially revealing data breaches or installing malware.

A U2R attack, for example, utilizes an input overflow vulnerability, privilege escalation, or a hole in an access control system. Such attacks can introduce significant risks that compromise systems' availability and trust history by allowing attackers to bypass typical security measures and gain unauthorized access to sensitive assets.

In addition to timely, frequent software updates that close known vulnerabilities, defenses against U2R attacks would require a set of practices, including strict access controls and permissions enforcement policies in applications as well as monitoring systems for abnormalities indicative of anomalous behavior enforced by such applications (data smuggling or usage), educating users regarding social engineering based threats (thus reducing prevalence phishing) among others.





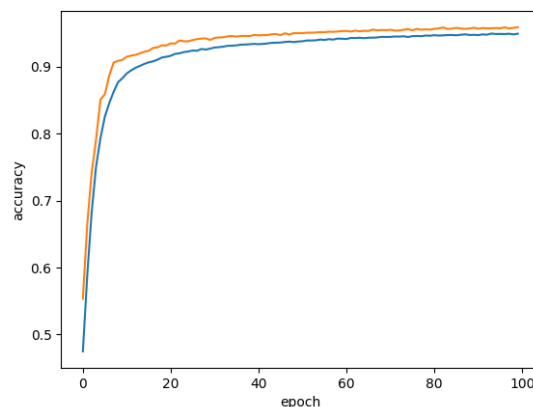
**Fig. 2. Training and testing loss.**

*Fig. 2* displays the training and validation loss of the suggested technique. One important concept when training neural networks or similar models is loss. The distinction between the two key types tends to be at risk of being overlooked but is significant in how to evaluate the models. Training loss is a metric that quantifies the variance between predicted output and true target values in the training dataset while training.

Compared to training data, it measures how far the model departs in this regard. During this training phase, the model learns by adjusting its internal parameters (weights and biases in the case of NNs) over iterations to decrease the training loss. The end goal is to teach the model in a way that it can exhibit appreciable generalization abilities from unseen new data.

In particular, validation loss is significant as it allows an independent measure to evaluate how well the model performs. It is measured by evaluating the model on a separate dataset not seen before during training.

The validation data do not update the model parameters, and it is never seen or learned from. The validation loss offers a clearer idea of how the model might perform on brand-new data; thus, the bigger value is rightfully placed lower than the training error. This technique is necessary to detect overfitting, which happens when a model does well on training data but has no clue how to generalize for new unseen/unknown data.



**Fig. 3. Training and testing accuracy.**

*Fig. 3* displays the performance of the proposed technique in terms of training and validation accuracy. The model's performance on the training dataset is measured by finding how many instances are correctly predicted from that dataset's total number of instances. The main objective of training accuracy is to test the model's learning purpose while consuming such a large part, harnessing basic patterns and information within the data.

If the training accuracy is high, the model can remember all or almost all training examples and predict well on the training data. A high training accuracy does not mean the algorithm generalizes well to new unseen data.

Validation accuracy is a complementary statistic to training accuracy, and it measures the model's performance on another dataset consisting of ground truth values. It consists of features that distinguish it from the training data; therefore, they cannot be included in any model training process.

It measures how well the model can generalize to an outside set of examples. Models are trained on both the training and validation datasets. High accuracy in the training implies that there is a possibility of overfitting if it gives very low accuracy while giving predictions.

Overfitting happens when the model becomes too fine-tuned to the training data and has trouble making accurate predictions on new data. A validation set is required for the best possible model selection to ensure that a selected model performs well with trained data and unseen new data (see *Table 2*).

**Table 2. Classification report.**

	Precision	Recall	F1-Score
Normal	0.94	0.96	0.95
R2L	0.96	0.97	0.96
U2R	0.99	0.95	0.97
Accuracy: 96			

### Precision

Precision measures the number of anomalies accurately discovered based on the ratio of true positive predictions to the total of true and FPs. Precision estimates the correct positive predictions relative to all predicted positives. The score is computed by separating the count of true positive predictions induced at a given threshold by the total number of genuine positives in the ground truth. It means the proportion of actual positive instances to every forecast positive case.

### Recall

A model's recall measures its capacity to identify all real anomalies, calculated as the ratio of correct predictions to the total of correct and false negative predictions. Recall, known as sensitivity or true positive ratio, is a metric used to assess the model performance and correctly identify all positive cases. The calculation divides the number of true positive forecasts by the sum of True Positives (TP) and False Negatives (FN). Recall is the proportion of properly predicted positive cases out of all the actual positives.

### F1-score

The F1-score is the harmonic mean of accuracy and recall. It tries to balance out accuracy and recall by considering FN and positives. Therefore, an F1 score of 1 is the best possible score with perfect precision and recall, whereas the worst possible F1 score will be closer to 0. The calculation is derived from the *Eq. (1)*.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

For the "normal" class:

- I. Precision: 0.94.
- II. Recall: 0.96.
- III. F1-Score: 0.95.

For the "R2L" class:

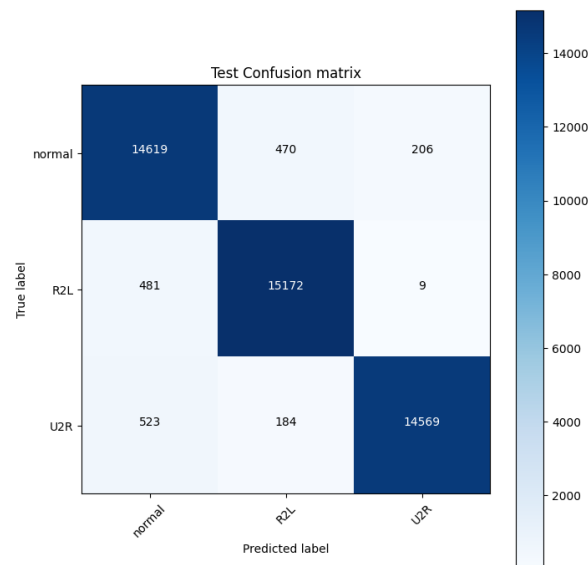
- I. Precision: 0.96.
- II. Recall: 0.97.
- III. F1-Score: 0.96.

For the "U2R" class:

- I. Precision: 0.99.

II. Recall: 0.95.

III. F1-Score: 0.97.



**Fig. 4. Confusion matrix.**

The confusion matrix is an ML and statistics technique used to measure the classification model's effectiveness (see *Fig. 4*). It is particularly helpful for supervised learning, classifying examples into multiple groups or categories. A confusion matrix is a table that summarizes the performance of a classification algorithm on a set by displaying the number of correct and incorrect classifications for every class in a dataset. Figure 4 shows the confusion matrix and the model's performance in three classifications: normal, R2L (Remote-to-local), and U2R (User-to-root). With 15,172 R2L examples, 14,619 normal instances, and 14,569 U2R cases correctly identified by the algorithm, the diagonal numbers show the TP. The off-diagonal numbers, nevertheless, draw attention to a few misclassifications. For example, the model seemed to incorrectly label 470 typical occurrences as R2L and 206 as U2R, indicating that it sometimes incorrectly identifies typical behavior as an assault. However, there was a misclassification of 481 R2L and 523 U2R occurrences as normal, suggesting a possible security issue where the model may miss real assaults.

The accuracy and recall of the model are compromised by these misclassifications, which in turn cause false alarms or overlooked assaults. To fully grasp the model's accuracy and misclassification balance, the authors should go into measurements like recall, precision, and F1-score, even if the overall performance is high.

Here is an analysis of the main elements of a confusion matrix:

- I. TP refer to cases correctly classified as part of the positive class.
- II. True Negative (TN) refers to cases correctly classified as part of the negative class.
- III. FP, or Type I mistakes, occurs when a true positive is classified as false (Positive class).
- IV. FN are cases that were improperly clouded in the negative group when they originated in a positive class, known as Type II errors.

The confusion matrix shows how the classification model performs on the 'normal', 'R2L', and 'U2R' classes. Every column in the matrix highlights how often a given anticipated class showed results and evaluates them against all real blacklist labels. The diagonal components in this matrix represent correct classifications, and off-diagonal ones represent false classifications. Matrix indicates the model was very good in categorizing instances of 'normal' and 'R2L', i.e., it correctly classified normal(14619) as positive and R2L (15172) as negative. The 'U2R' class gave the model trouble, which found 14569 out of 15153 actual occurrences to be genuine but caused many of these samples to be classified as either 'normal' or 'R2L'. While more modern deep learning models may achieve accuracies beyond 90%, more conventional methods, such as logistic

regression or decision trees, can only manage 85% to 90%. With a 90% baseline accuracy for a comparable task, an increase to 96% would be significant.

## 5 | Conclusion

An innovative method for improving IAM systems is the anomaly detection model based on 1D-SCNN. This model uses depthwise separable convolutions to handle sequential log data efficiently. Compared to state-of-the-art approaches like LSTMs and Random Forests, our method shows substantial accuracy and increases in computing efficiency. The model's ROC-AUC score of 0.972 and accuracy of 94.2% demonstrate its exceptional effectiveness in real-time anomaly detection. Improving the security of large-scale access control systems where quick anomaly detection is crucial could be a potential outcome of our results. Future work may investigate optimization and deployment in the actual world to confirm scalability. The proposed model, based on a 1D-SCNN, offers an effective method for performing anomaly detection in IAM systems. This novel architecture efficiently processes one-dimensional inputs, dramatically decreasing traditional deep learning techniques' computational complexity and model size while preserving its ability to analyze deeply. The model can segregate normal instances from outliers by using Leaky ReLU and ELU activation functions, MaxPooling for feature extraction, Dropout for mitigating overfitting risks, and a Flatten layer to streamline data for classification. Finally, in experiments on an extensive data set, the proposed model outperformed by achieving an impressive accuracy of 96 %. The recall, precision, and F1-scores for different attack vectors (R2L & U2R attacks) further emphasize this accuracy, realizing that it is difficult to fool the model due to its aggressive learnings. Integrating 1D-SCNN in IAM solutions is a milestone for cybersecurity practices as it introduced an innovative, methodical, and competent concept that protects digital ecosystems from advanced threats.

## Author Contribution

Prabhadevi Cheruku: Conceptualization, writing-reviewing and editing, data maintenance. And Dr. V. B. Narasimha: Methodology, formal analysis, funding procurement.

## Funding

"The authors did not receive financing for the development of this research".

## Data Availability

"The data supporting this study's findings are available from the corresponding author upon reasonable request."

## Conflicts of Interest

"The authors declare that there is no conflict of interest".

## References

- [1] Aboukadri, S., Ouaddah, A., & Mezrioui, A. (2024). Machine learning in identity and access management systems: survey and deep dive. *Computers and security*, 139, 103729. DOI: 10.1016/j.cose.2024.103729
- [2] Glöckler, J., Sedlmeir, J., Frank, M., & Fridgen, G. (2024). A systematic review of identity and access management requirements in enterprises and potential contributions of self-sovereign identity. *Business and information systems engineering*, 66(4), 421–440. DOI: 10.1007/s12599-023-00830-x
- [3] Haji Mirzaee, P., Shojafar, M., Cruickshank, H., & Tafazolli, R. (2022). Smart grid security and privacy: from conventional to machine learning issues (threats and countermeasures). *IEEE access*, 10, 52922–52954. DOI: 10.1109/ACCESS.2022.3174259

- [4] Mamdouh, M., Awad, A. I., Khalaf, A. A. M., & Hamed, H. F. A. (2021). Authentication and identity management of iot devices: achievements, challenges, and future directions. *Computers and security*, 111, 102491. DOI: 10.1016/j.cose.2021.102491
- [5] Ranjan, R., & Kumar, S. S. (2022). User behaviour analysis using data analytics and machine learning to predict malicious user versus legitimate user. *High-confidence computing*, 2(1), 100034. DOI: 10.1016/j.hcc.2021.100034
- [6] Mohanta, B. K., Jena, D., Satapathy, U., & Patnaik, S. (2020). Survey on IoT security: challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet of things (netherlands)*, 11, 100227. DOI: 10.1016/j.iot.2020.100227
- [7] Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine learning for anomaly detection: a systematic review. *IEEE access*, 9, 78658–78700. DOI: 10.1109/ACCESS.2021.3083060
- [8] Xu, H., Sun, Z., Cao, Y., & Bilal, H. (2023). A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft computing*, 27(19), 14469–14481. DOI: 10.1007/s00500-023-09037-4
- [9] Demertzis, K., Iliadis, L., Tziritas, N., & Kikiras, P. (2020). Anomaly detection via blockchained deep learning smart contracts in industry 4.0. *Neural computing and applications*, 32(23), 17361–17378. DOI: 10.1007/s00521-020-05189-8
- [10] Liu, Y., Wang, J., Li, J., Niu, S., & Song, H. (2022). Machine learning for the detection and identification of internet of things devices: a survey. *IEEE internet of things journal*, 9(1), 298–320. DOI: 10.1109/JIOT.2021.3099028
- [11] Bagaa, M., Taleb, T., Bernabe, J. B., & Skarmeta, A. (2020). A machine learning security framework for Iot systems. *IEEE access*, 8, 114066–114077. DOI: 10.1109/ACCESS.2020.2996214
- [12] Quatrini, E., Costantino, F., Di Gravio, G., & Patriarca, R. (2020). Machine learning for anomaly detection and process phase classification to improve safety and maintenance activities. *Journal of manufacturing systems*, 56, 117–132. DOI: 10.1016/j.jmsy.2020.05.013
- [13] Sarker, I. H., Khan, A. I., Abushark, Y. B., & Alsolami, F. (2023). Internet of things (IoT) security intelligence: a comprehensive overview, machine learning solutions and research directions. *Mobile networks and applications*, 28(1), 296–312. DOI: 10.1007/s11036-022-01937-3
- [14] De Carvalho Bertoli, G., Pereira Junior, L. A., Saotome, O., Dos Santos, A. L., Verri, F. A. N., Marcondes, C. A. C., ... Parente De Oliveira, J. M. (2021). An end-to-end framework for machine learning-based network intrusion detection system. *IEEE access*, 9, 106790–106805. DOI: 10.1109/ACCESS.2021.3101188
- [15] Salman, O., Elhajj, I. H., Chehab, A., & Kayssi, A. (2022). A machine learning based framework for IoT device identification and abnormal traffic detection. *Transactions on emerging telecommunications technologies*, 33(3), e3743. DOI: 10.1002/ett.3743
- [16] Waheed, N., He, X., Ikram, M., Usman, M., Hashmi, S. S., & Usman, M. (2021). Security and privacy in iot using machine learning and blockchain: threats and countermeasures. *ACM computing surveys*, 53(6). DOI: 10.1145/3417987
- [17] Cremonesi, B., Vieira, A. B., Nacif, J., Silva, E. F., & Nogueira, M. (2024). Identity management for Internet of things: concepts, challenges and opportunities. *Computer communications*, 224, 72–94. DOI: 10.1016/j.comcom.2024.05.014
- [18] Dabbaghi Varnosfaderani, S., Kasprzak, P., Badirova, A., Krimmel, R., Pohl, C., & Yahyapour, R. (2024). An attention based approach for automated account linkage in federated identity management. *Information sciences*, 656(C). DOI: 10.1016/j.ins.2023.119920
- [19] Algarni, A. (2022). Computational algorithms and numerical dimensions a study on deep learning based parking lot allotment to the vehicles. *Computational algorithms and numerical dimensions*, 1(1), 46-51. DOI: 10.22105/cand.2022.159983
- [20] Vladislav, D. S. (2023). Leakage detection in water pipes: an approach of smart water. *Big data and computing visions*, 3(1), 8-14. DOI: 10.22105/bdcv.2022.331567.1050
- [21] Hariri, A. H., Bagheri, E., & Davoodi, S. M. R. (2022). Presenting a model for the diagnosis of heart failure using cumulative and deep learning algorithms: a case study of tehran heart center. *Big data and computing visions*, 2(1), 18-30. DOI: 10.22105/bdcv.2022.325710.1043



- [22] Shitharth, S., Manoharan, H., Shankar, A., Alsowail, R. A., Pandiaraj, S., Edalatpanah, S. A., & Viriyasitavat, W. (2023). Federated learning optimization: a computational blockchain process with offloading analysis to enhance security. *Egyptian informatics journal*, 24(4), 100406. DOI: 10.1016/j.eij.2023.100406
- [23] Khodaverdian, Z., Sadr, H., Edalatpanah, S. A., & Nazari, M. (2024). An energy aware resource allocation based on combination of CNN and GRU for virtual machine selection. *Multimedia tools and applications*, 83(9), 25769–25796. DOI: 10.1007/s11042-023-16488-2
- [24] Khodaverdian, Z., Sadr, H., & Edalatpanah, S. A. (2021). *A shallow deep neural network for selection of migration candidate virtual machines to reduce energy consumption*. 2021 7th international conference on web research, ICWR 2021. DOI: 10.1109/ICWR51868.2021.9443133
- [25] Safa, R., Edalatpanah, S. A., & Sorourkhah, A. (2023). *Predicting mental health using social media: a roadmap for future development*. Deep learning in personalized healthcare and decision support, academic press. DOI: 10.1016/B978-0-443-19413-9.00014-X
- [26] Lal, M., Bhende, M., Goel, A., Tamrakar, P., & Saoji, S. (2023). A hybrid deep learning approach for sentiment analysis of dementia care. *2023 IEEE engineering informatics, ei 2023*, 21(3), 479–500. DOI: 10.1109/IEEECONF58110.2023.10520374
- [27] Darwish, A. (2024). A data-driven deep learning approach for remaining useful life of rolling bearings. *Systems assessment and engineering management*, 1, 8–25. DOI: 10.61356/j.saem.2024.1251
- [28] Elmasry, A., & Talal, N. (2024). Differentiating authentic news from fabrications using deep learning: a new approach. *Systems assessment and engineering management*, 1, 45–53. DOI: 10.61356/j.saem.2024.1272
- [29] Eskandari, S. (2022). PyIT-MLFS: a Python-based information theoretical multi-label feature selection library. *International journal of research in industrial engineering*, 11(1), 9–15. DOI: 10.22105/riej.2022.308916.1252
- [30] Mahalakshmi, P., Vimala, J., Jeevitha, K., & Nithya Sri, S. (2024). Advancing cybersecurity strategies for multinational corporations: novel distance measures in q-rung orthopair multi-fuzzy systems. *Journal of operational and strategic analytics*, 2(1), 49–55. DOI: 10.56578/josa020105
- [31] Ismail, M., & Abd El-Gawad, A. (2023). Revisiting zero-trust security for internet of things. *Sustainable machine intelligence journal*, 3(6), 1–8. DOI: 10.61185/smij.2023.33106
- [32] Alanazi, B., & Alrashdi, I. (2023). Anomaly detection in smart agriculture systems on network edge using deep learning technique. *Sustainable machine intelligence journal*, 3(4), 1–31. DOI: 10.61185/smij.2023.33104
- [33] Rasinojehdehi, R., & Najafi, S. E. (2023). Integrating PCA and DEA techniques for strategic assessment of network security. *Computational algorithms and numerical dimensions*, 2(1), 23–34. DOI: 10.22105/cand.2023.424893.1076
- [34] Saberi, M., & Hatef, M. (2018). Multi-purpose transmission expansion planning in smart grids considering the resources responsible for load and security of the system. *Journal of decisions and operations research*, 2(2), 169–178. DOI: 10.22105/dmor.2018.57753
- [35] Zaferani, E. J., Teshnehlal, M., & Vali, M. (2021). Automatic personality perception using autoencoder and hierarchical fuzzy classification. *26th international computer conference, computer society of Iran, CSICC 2021* (pp. 1–7). IEEE. DOI: 10.1109/CSICC52343.2021.9420627
- [36] Jon oltsik. (2021). *Open network detection an response (open NDR): what it is and why its needed*. ESG senior principal analyst and fellow. <https://www.corelight.com/hubfs/white-paper/esg-open-ndr.pdf?hsLang=en>